The Startup Drake Equation

The Drake explanation · The Startup Drake Equation · Overwhelming · Design out risk · Founding team · Design out weaknesses · Play asymmetric games · High-risk first

Most startups fail, even when the founders are smart, driven, passionate, capable, and are solving a problem that at least a few people would pay to have solved. Why?

We already explored the primary causes of startup failure[2] and how to combat those.

Since failure is the primary result, however, perhaps "why do startups fail" is the wrong question. Startups fail by default; the question is: Why do they ever survive?

And: Why *specifically* are they default-dead? Is there something we can learn from that?

"Things went from bad to worse, but we're hopeful now that we're doing badly again."

credit[1]

THE DRAKE EXPLANATION

Frank Drake created his eponymous Equation[3] in 1961 to guide discussions at the first meeting of SETI (the Search for Extra-Terrestrial Life). It became a famous a way of estimating how many alien civilizations we should expect to see in the night sky:

$$N = R_* \cdot f_\mathrm{p} \cdot n_\mathrm{e} \cdot f_\mathrm{l} \cdot f_\mathrm{i} \cdot f_\mathrm{c} \cdot L$$

Frank Drake, Cornell, 2017 credit[4]

In English: There are billions of stars (R_*), some fraction of which have planets (f_p), some fraction of which are habitable (n_e), and sometimes time life actually forms (f_l), and even becomes intelligent (f_i), and transmits detectable signals into space (f_c), and have been doing so for long enough for us to see it (L). So that's how many aliens civilizations we should see (N).

Or in modern Marketing language:[5]

> The galaxy produces many warm leads, but every step of the conversion funnel is brutally leaky, so it's hard to convert a sale.

Of course the most salient fact about detecting alien civilizations is: We haven't detected any alien civilizations. We know from direct observation that the first few values of the Drake Equation are astronomically large. So that means one or perhaps all of those conversion steps are vanishingly rare.

Startups feel like this too. Countless side projects are started each day, some fraction of which are intended to become money-making endeavors. While the success rate isn't as low as alien civilizations apparently are, perhaps 999 out of 1000[6] fail the chain of probabilities—failing to create a venture where the owner quits their day job and brags to outwardly-supportive-but-inwardly-jealous Twitter "followers" about achieving Product/Market Fit.[7]

Startups face a chain of risks, or as I like to say, a chain of "ands"[8]—many things all have to go right. Of course "all things" rarely go right simultaneously; this is why startups typically fail.

credit[9]

THE STARTUP DRAKE EQUATION

Here are just some of the factors in the Startup Drake Equation, the failure of any one of which is terminal:

- Product that people actually do want to pay for
- Able to get those people's attention amidst the noise of the Internet
- Pricing that those people will accept (and that is greater than your costs)
- Competitive and distinctive enough to be chosen among the crowd
- Develop a repeatable and profitable customer acquisition strategy
- Able to build the product that was promised by the home page
- Able to sustain the delivery of value over time so customers stay and keep paying
- Able to fund the venture, either through early profits or fundraising
- Able to work well with co-founders, and not break up (or able do it all by yourself)
- Able to attract and retain talent

- Able to psychologically handle many years of deep effort and stress and pain
- Get lucky here and there

It's easy to find examples of failures due to each factor. A classic example is the non-technical founder who unsuccessfully outsources the product to a consultant (fails "can build it"). Another is the technical founder who builds forever without talking to customers, and as a result builds the wrong thing (fails "product that people want to pay for" and "able to get attention"). Another is the "I had the problem myself, so I built it" origin story, but it turns out not enough other people have the problem and the budget and desire to have it solved. Another is that it was too hard to push through the pain of iteration and pivoting, and the day job pays well and there's a two-year-old at home, so after six months the founder gives up.

The insights of this model are:

1. The failure of just one category is fatal, so we should spend more time identifying and addressing the biggest risk-areas.
2. We can think in terms of "reducing risk" or "increasing chances" rather than "best ideas" or "unique strategy" or "changing the world".

Here's how to use these insights to increase the chance of success.

COMPLETELY CRUSH A FEW AREAS TO OVERCOME OTHER RISKS AND WEAKNESSES

No organization will be low-risk on all areas. But perhaps some areas can be 100%, or effectively "greater," and making up for other deficiencies.

For example, a top-1% engineer might satisfy the question of whether we can build it, but "greater" is wrapping a strategy around the founder of a successful open source project, which then becomes a unique competitive and marketing advantage, which overcome deficiencies like not having unique features in the product or not having special skills in advertising. Or, if you're an renowned expert in some market, that decreases market risk, both because that's a marketing advantage and because you probably have insights that others lack.* Whereas if you're entering a market you know nothing about, your education might prove fatal.

I give several examples of this in The Important Thing.[11] You gain both focus and a higher probability of success when you have a singular winning attribute, even when you're terrible at everything else.[12] Singular focus and differentiation can overcome deficiencies elsewhere in the Startup Drake Equation:

> Even better than "different," is to be extreme in that difference. Not just a minimal UI, but so minimal it works on the command-line. Not just great design, but so remarkable people buy it only for that, and it's written up on designer's blogs. Not just a new algorithm that solves an old problem, but one that uncovers new things that no one else does, even at the expense of missing things that others catch. You can't do this for all aspects of your product and business —indeed, even a single one is already powerful—but extremity is how you maximize the power of the few things that make you special.

---

* Although be wary; your experience could be blinding you.[10]

"The rejection rate for this kind of organ transplant is quite high, so we're going to cram three or four in there to better our odds."

credit[13]

CHOOSE PRODUCT/MARKET/CUSTOMER TO DECREASE RISK

We typically think about "target market" with a success-oriented question like: "Who would be delighted by this product?" But a risk-oriented version of this question is sometimes easier to answer:

What would be the *easiest* customer segment for us to target?

Here "easy" doesn't mean just mean that you can, or that it's fun, but also that it is profitable. As a negative example, indie hackers* enjoy selling to their peer indie hackers, but indie hackers have no money

* Found mainly on Twitter, the quintessential "indie hacker" is a solo founder, who wants no employees, who can build software without assistance, who values freedom, flexibility, and autonomy over money or prestige. They therefore build simple (but hopefully delightful) products, at low prices, designed to be profitable without scaling.

and usually go out of business in less than a year, so that's a very risky segment. Instead, stable small businesses like dentist offices have large budgets, rarely change software, and last for decades; this is a better market.* Large, growing markets are better still, because "large" means there's lots of niches[14] in which to get started, and lots of adjacencies[15] to expand into later, and incumbents[16] are fighting over new customers, not focussed on new entrants who aren't big enough yet to be worrisome. Big, growing markets are easier.

The other key factor is in the phrase "for us". An Enterprise segment of a market might be terrific in terms of size and willingness-to-pay-gobs-of-money-for-a-decade, but a new company run by a single person will not be able to deliver the complex software, integrations, governance protocols, and professional services that even one Enterprise customer demands, therefore this would be a risky market *for them*.

You must select low-risk markets, which are easy for you to address. You could pick a different word than "easy"—lucrative, growing, profitable—but I like "easy" because it keeps things personal. Do you think it will be easy? If so, you're wrong—it will be harder than you think, but still possible. Whereas if you already think it will be difficult, it's also worse than you fear: impossible.

FOUNDING TEAM

When a co-founder is a top-1% engineer, there's almost no risk that the company will fail only because the product couldn't be built.** When a co-founder is a top-1% growth marketer, and if the market exists, there's

* Better for sustainable revenue, but difficult to sell, as dentists are hard to get on the phone, and aren't usually in the market for new software. Nothing is easy!

** Except in AI,[17] where the normal rules are different, and the world's best engineers often in fact aren't making it work.

almost no risk that the company will fail only because it was impossible to get attention.

This is one reason why investors like two founders—"one to build it, one to get rid of it" as we used to say at ITWatchDogs. It's not just "getting twice the work done," and not just "someone to commiserate with," but also because you dramatically improve at least two of the variables in the Startup Drake equation.

DESIGN OUT WEAKNESSES

You have weaknesses,[18] which increases risk across the Startup Drake equation. But, you can make choices that slalom around most of your weaknesses irrelevant, thus also removing the risks.

If you're creating a startup on the side, while you hold a day job and a two-year-old, then you should serve an audience who doesn't want tech support, or at least is fine with support tickets taking 48 hours for a response. That might have implications on how complex the product is, how intuitively it is designed, what customers expect of it, and its price.[19] Rather than seeing those as negative constraints, instead realize that these are a superior strategy and product, because it avoids a weakness (amount and consistency of time-available). Indeed, besides solving for the weakness, there are benefits: Your profit margin is higher (because you don't have support costs) and you can sell around the world (because neither language nor timezones are a barrier) and it will be delightful to use (so the only communication is asynchronous praise on Reddit). This can even be done at scale (GMail, Facebook, Twitter, most hardware products). Suddenly a negative "constraint" looks like an insightful advantage.

Or if you're a terrible designer like me,[20] it would be high-risk to make a product that must appeal to designers or marketing agencies, i.e. people

IF ENGINEERS WERE TREATED LIKE DESIGNERS

credit[21]

who value and appreciate great design. You'll do just fine with infrastructure engineers or backend enterprise systems.

Or if you're terrible at marketing, you could create a collaborative product where people have to invite other people in order to use it. While that mechanism is difficult to get started, it means even poor marketing can result in a growing, healthy company. (It's funny that a "viral"[22] company is also "healthy.")

Or if you cannot write code, but you are good at selling yourself and solving a class of valuable problems, you could avoid the world of software (whether as-a-service or not), instead creating a "productized service," in which you sell services, but fulfill that service at low internal cost (and therefore high profit) thanks to your "secret sauce" internal workflows, spreadsheets, and no-code software, which has to be good enough only for your own employees to use.

By listing your weaknesses as fervently as you list your strengths, you can increase the chance of success by avoiding them, and embracing the knock-on implications.

I cover this in more detail in my article on Pivot Points.[18]

PLAY ASYMMETRIC GAMES

Any given activity might work, or might not. Even the probability of success is unknown—the variables in the Startup Drake Equation don't have clear values.[23] Although that means there will be failures, it also suggests there will be some successes. If the positive magnitude of the few successes exceed the negative magnitude of the failures, we come out as a success overall.

After all, in most startups, most things are going pretty poorly. New people joining the company often say, "Wow, I can't believe you're still doing _____ and it's still working!" They aren't wrong; they're accidentally pointing out that some things can be so powerfully positive, that it overwhelms deficiencies.

Many examples of this appear in the "Asymmetric" section of What makes a strategy great.[24] It's summarized best by Jeff Bezos, who led a company that never stopped taking asymmetric bets:

> "I've made billions of dollars of failures at Amazon.com. Literally.
>
> But a few big successes compensate for dozens and dozens of things that didn't work."
> —Jeff Bezos, The Guardian,[25] 2014

On a smaller scale, there are marketing efforts with a fixed cost, and also a fixed maximum upside, like advertisement, where you pay whether or not someone signs up. Or there are marketing efforts with a fixed cost, that could generate new customers for years to come, as with reputation (social media, community forums) or media (successful SEO).

Or there are markets where there are many different ways to win, many niches to try, many ways to expand from a successful foothold into a wider space (i.e. the "large and growing global" markets), and markets which are stagnant, perhaps shrinking, where every sale is a hard-scrabble against desperate but monied incumbents.

Pick games where the downside is 1x but the upside potential is 10x or 100x. Then if the realized benefit is "only" 3x, the game is still a success, and hopefully makes up for other weaknesses.

ATTACK THE HIGH-RISK AREAS FIRST

With complex projects, the Risk-First Heuristic[26] tells us to tackle the high-risk things first.

If you start with low-risk tasks, you'll surely succeed, but the high-risk tasks remain unresolved, and remain high-risk. Nine months later, when you finally tackle the high-risk areas, you might discover the entire venture is unworkable. Even if it's not fatal, new insights means rework, which means you've wasted time. It's smarter to address the risky things first, learn from them, change because of them, and complete the rest of the project informed by them.

The Startup Drake Equation lists potential risk areas. Even after designing our product, market, and strategy to circumvent or remove risks, some high-risk areas will remain. Circle those, and address them first.

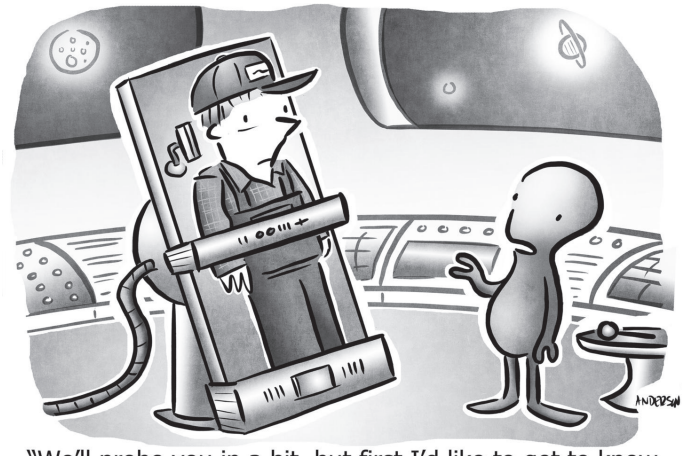
So if we're incorporating AI[17] into the product, make sure that works first, because usually it doesn't (at least not as of this writing, as thousands of self-styled "AI Startups" have discovered). Or if we're unsure what potential customers would actually pay for (as often they *have* a problem but don't want to pay to solve it), we should find that out[27] before writing code. Or if we've never marketed a new product before, we should make

advertisements and landing pages that lead to a waiting list, to make sure we can garner attention before building something that no one will ever see.[28]

All startups are risky, and even with this model in hand, most startups will still fail.

But, by clearly articulating the risks, reducing some risks through the founding team and selecting the market and product that is right *for us*, solving one or two so well that they overwhelm other risks, constructing a strategy that designs around things that would otherwise be weaknesses, and attacking the remaining high-risk areas first, you can dramatically improve the chance of success.

Those aliens are out there!



"We'll probe you in a bit, but first I'd like to get to know the real you. Your hopes, your goals, your dreams… Who is Fred Wilson? What makes him tick? And *then* the probing. Lots and lots of probing."

credit[29]

The current version of this article:
`https://asmartbear.com/startup-drake-equation/`

More articles & socials:
`https://asmartbear.com`

References

1. https://andertoons.com/chart/cartoon/3709/things-went-from-bad-to-worse-but-were-hopeful-now-that-were-doing-badly-again
2. https://longform.asmartbear.com/avoid-blundering/
3. https://en.wikipedia.org/wiki/Drake_equation#
4. https://en.wikipedia.org/wiki/Frank_Drake
5. https://longform.asmartbear.com/the-leading-provider/
6. https://twitter.com/asmartbear/status/1820192235981332735
7. https://longform.asmartbear.com/product-market-fit/
8. https://longform.asmartbear.com/capturing-luck/
9. https://andertoons.com/smartphone/cartoon/8362/hey-said-take-us-to-leader-hello-aliens
10. https://longform.asmartbear.com/expert-harmful/
11. https://longform.asmartbear.com/the-important-thing/
12. https://longform.asmartbear.com/worse-but-unique/
13. https://andertoons.com/hospital/cartoon/7349/rejection-rate-organ-trans-plant-high-cram-three-or-four-in
14. https://longform.asmartbear.com/icp-ideal-customer-persona/
15. https://longform.asmartbear.com/adjacency/
16. https://longform.asmartbear.com/startup-beats-incumbent/
17. https://longform.asmartbear.com/ai-startups/
18. https://longform.asmartbear.com/pivot-points/
19. https://longform.asmartbear.com/pricing-determines-your-business-model/
20. https://longform.asmartbear.com/design/
21. https://www.reddit.com/r/Design/comments/3latn6/if_engineers_were_treated_like_designers/
22. https://longform.asmartbear.com/going-viral/
23. https://longform.asmartbear.com/probability-words/
24. https://longform.asmartbear.com/great-strategy/#asymmetric
25. https://www.theguardian.com/technology/2014/dec/03/jeff-bezos-billions-dollars-failures-amazon?utm_source=longform.asmartbear.com&utm_campaign=longform.asmartbear.com&utm_medium=post
26. https://riskfirst.org/
27. https://longform.asmartbear.com/customer-development/
28. https://longform.asmartbear.com/put-down-the-compiler/
29. https://andertoons.com/aliens/cartoon/7878/well-probe-you-but-first-id-like-to-get-to-know-real-you